

On the implementability of Global Types

Gabriele Genovese

Supervisor: Cinzia Di Giusto

4 July 2025

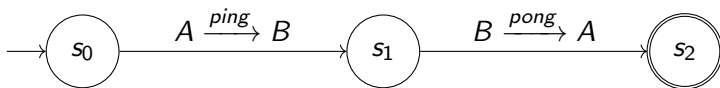


Formal Distributed Systems

- ▶ Abstractions used to simplify the study and development:
 - ▶ Global Types
 - ▶ Message Sequence Charts (MSC)
- ▶ Useful for proving various properties

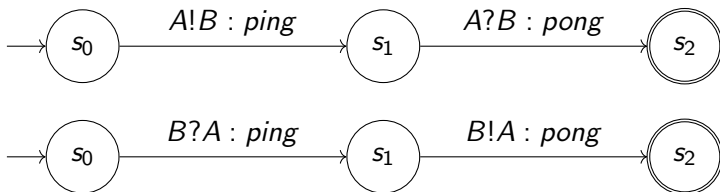
Global Types

- ▶ Description of a **global** behavior of a communication system.
- ▶ Automaton: visual representation of the global type.



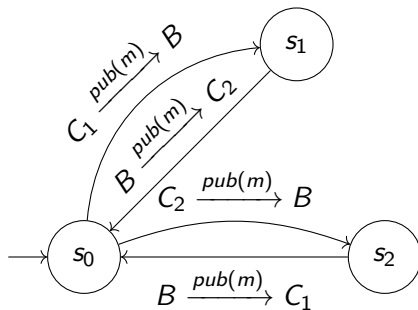
Local type

- ▶ Point of view of a participant
- ▶ Typically obtained with a *projection* operation



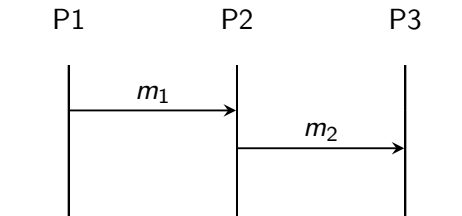
Example

Message Queuing Telemetry Transport (MQTT) protocol with two clients.



Message Sequence Charts (MSC)

Diagrams used to represent traces of a behavior of the system.



Events: send m_1 , receive m_1 , send m_2 , receive m_2

The implementability problem

Property to guarantee: **respectfulness** the behavior described.

Does the implementation of a system **respects** the behavior described?

Example

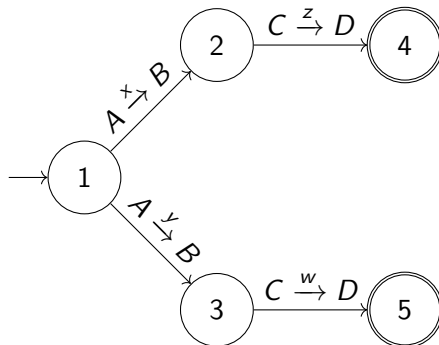
- ▶ 4 participants: A, B, C, D
- ▶ 4 messages: x, y, z, w
- ▶ Specification description:

```
A send B either message x or y.
```

```
if A send B message x,  
    then C send B message z.
```

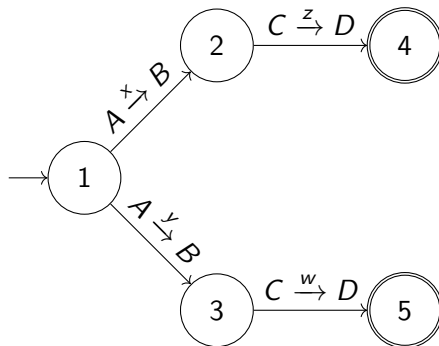
```
if A send B message y,  
    then C send B message w.
```


Example's global type



Example's global type

This Global Type is **not** implementable because c doesn't know what b received.



Reduction to sync

A global type G is implementable in **p2p** iff:

1. $L_{\text{p2p}}(\text{proj}(G))$ is a set of sync MSCs;
2. $\text{proj}(G)$ is orphan-free in p2p;
3. $L_{\text{p2p}}(\text{proj}(G))$ is deadlock-free
4. G is implementable in sync

Reduction to sync

A global type G is implementable in **p2p** iff:

1. $L_{p2p}(proj(G))$ is a set of sync MSCs;
2. $proj(G)$ is orphan-free in p2p;
3. $L_{p2p}(proj(G))$ is deadlock-free
4. **G is implementable in sync** ←

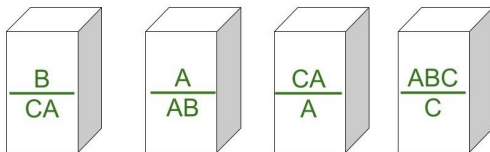
My contribution

Extension of a proof to a general framework:

- ▶ Original theorem: checking implementability for bounded MSCs is undecidable
- ▶ **Now**: checking implementability for sync-global types is undecidable
- ▶ Proof: by reduction to the PCP problem

Post Correspondance Problem (PCP)

Given a set of tiles, find an ordering such that the strings formed by the top and bottom halves are equal.



Future work

- ▶ Extend existing results using the com-general framework
- ▶ Extend and adapt a model checking tool to the com-general framework

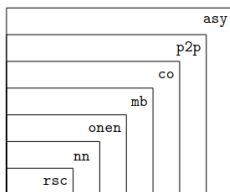
Reduction to sync

A global type G is implementable in **p2p** iff:

1. $L_{\text{p2p}}(\text{proj}(G))$ is a set of sync MSCs;
2. $\text{proj}(G)$ is orphan-free in p2p;
3. $L_{\text{p2p}}(\text{proj}(G))$ is deadlock-free
4. G is implementable in sync

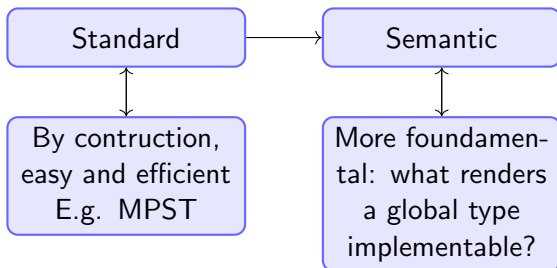
Hierarchy of communication models

More interesting: async, p2p, mb (mailbox), rsc (sync).



State of the art

The study about implementability can be summarized in:



Our approach: semantic

- ▶ What render a specification implementable?
- ▶ What is the limit? Why syntactical constraints works?
- ▶ **Aim:** Extend existing results and generalize to different communication models

Conclusion & Future work

Summerize:

- ▶ Study of the implementability problem for global types
- ▶ Proof of undecidability
- ▶ Extend existing results and tool using the com-general framework

Other activities

- ▶ Participation at an International **Conference**, DisCoTec
- ▶ Obtained a **PhD grant** (DS4H)
- ▶ Future participation to a **Summer School** in Software Verification in Edimburgh

Thanks! Questions?